[ffmpeg命令详解(转)](http://www.cnblogs.com/AllenChou/p/7048528.html)

FFmpeg是一套可以用来记录、转换数字音频、视频，并能将其转化为流的开源计算机程序。采用LGPL或GPL许可证。它提供了录制、转换以及流化音视频的完整解决方案。它包含了非常先进的音频/视频编解码库libavcodec，为了保证高可移植性和编解码质量，libavcodec里很多code都是从头开发的。[百度百科]

**ffmpeg使用语法**

ffmpeg使用语法：

ffmpeg [[options][`-i' input\_file]]... {[options] output\_file}...

如果没有输入文件，那么视音频捕捉就会起作用。

作为通用的规则，选项一般用于下一个特定的文件。如果你给 –b 64选项，改选会设置下一个视频速率。对于原始输入文件，格式选项可能是需要的。

缺省情况下，ffmpeg试图尽可能的无损转换，采用与输入同样的音频视频参数来输出。

3．选项

a) 通用选项

-L license

-h 帮助

-fromats 显示可用的格式，编解码的，协议的。。。

-f fmt 强迫采用格式fmt

-I filename 输入文件

-y 覆盖输出文件

-t duration 设置纪录时间 hh:mm:ss[.xxx]格式的记录时间也支持

-ss position 搜索到指定的时间 [-]hh:mm:ss[.xxx]的格式也支持

-title string 设置标题

-author string 设置作者

-copyright string 设置版权

-comment string 设置评论

-target type 设置目标文件类型(vcd,svcd,dvd) 所有的格式选项（比特率，编解码以及缓冲区大小）自动设置，只需要输入如下的就可以了：  
ffmpeg -i myfile.avi -target vcd /tmp/vcd.mpg

-hq 激活高质量设置

-itsoffset offset 设置以秒为基准的时间偏移，该选项影响所有后面的输入文件。该偏移被加到输入文件的时戳，定义一个正偏移意味着相应的流被延迟了 offset秒。 [-]hh:mm:ss[.xxx]的格式也支持

b) 视频选项

-b bitrate 设置比特率，缺省200kb/s

-r fps 设置帧频 缺省25

-s size 设置帧大小 格式为WXH 缺省160X128.下面的简写也可以直接使用：  
Sqcif 128X96 qcif 176X144 cif 252X288 4cif 704X576

-aspect aspect 设置横纵比 4:3 16:9 或 1.3333 1.7777

-croptop size 设置顶部切除带大小 像素单位

-cropbottom size –cropleft size –cropright size

-padtop size 设置顶部补齐的大小 像素单位

-padbottom size –padleft size –padright size –padcolor color 设置补齐条颜色(hex,6个16进制的数，红:绿:兰排列，比如 000000代表黑色)

-vn 不做视频记录

-bt tolerance 设置视频码率容忍度kbit/s

-maxrate bitrate设置最大视频码率容忍度

-minrate bitreate 设置最小视频码率容忍度

-bufsize size 设置码率控制缓冲区大小

-vcodec codec 强制使用codec编解码方式。如果用copy表示原始编解码数据必须被拷贝。

-sameq 使用同样视频质量作为源（VBR）

-pass n 选择处理遍数（1或者2）。两遍编码非常有用。第一遍生成统计信息，第二遍生成精确的请求的码率

-passlogfile file 选择两遍的纪录文件名为file

c)高级视频选项

-g gop\_size 设置图像组大小

-intra 仅适用帧内编码

-qscale q 使用固定的视频量化标度(VBR)

-qmin q 最小视频量化标度(VBR)

-qmax q 最大视频量化标度(VBR)

-qdiff q 量化标度间最大偏差 (VBR)

-qblur blur 视频量化标度柔化(VBR)

-qcomp compression 视频量化标度压缩(VBR)

-rc\_init\_cplx complexity 一遍编码的初始复杂度

-b\_qfactor factor 在p和b帧间的qp因子

-i\_qfactor factor 在p和i帧间的qp因子

-b\_qoffset offset 在p和b帧间的qp偏差

-i\_qoffset offset 在p和i帧间的qp偏差

-rc\_eq equation 设置码率控制方程 默认tex^qComp

-rc\_override override 特定间隔下的速率控制重载

-me method 设置运动估计的方法 可用方法有 zero phods log x1 epzs(缺省) full

-dct\_algo algo 设置dct的算法 可用的有 0 FF\_DCT\_AUTO 缺省的DCT 1 FF\_DCT\_FASTINT 2 FF\_DCT\_INT 3 FF\_DCT\_MMX 4 FF\_DCT\_MLIB 5 FF\_DCT\_ALTIVEC

-idct\_algo algo 设置idct算法。可用的有 0 FF\_IDCT\_AUTO 缺省的IDCT 1 FF\_IDCT\_INT 2 FF\_IDCT\_SIMPLE 3 FF\_IDCT\_SIMPLEMMX 4 FF\_IDCT\_LIBMPEG2MMX 5 FF\_IDCT\_PS2 6 FF\_IDCT\_MLIB 7 FF\_IDCT\_ARM 8 FF\_IDCT\_ALTIVEC 9 FF\_IDCT\_SH4 10 FF\_IDCT\_SIMPLEARM

-er n 设置错误残留为n 1 FF\_ER\_CAREFULL 缺省 2 FF\_ER\_COMPLIANT 3 FF\_ER\_AGGRESSIVE 4 FF\_ER\_VERY\_AGGRESSIVE

-ec bit\_mask 设置错误掩蔽为bit\_mask,该值为如下值的位掩码 1 FF\_EC\_GUESS\_MVS (default=enabled) 2 FF\_EC\_DEBLOCK (default=enabled)

-bf frames 使用frames B 帧，支持mpeg1,mpeg2,mpeg4

-mbd mode 宏块决策 0 FF\_MB\_DECISION\_SIMPLE 使用mb\_cmp 1 FF\_MB\_DECISION\_BITS 2 FF\_MB\_DECISION\_RD

-4mv 使用4个运动矢量 仅用于mpeg4

-part 使用数据划分 仅用于mpeg4

-bug param 绕过没有被自动监测到编码器的问题

-strict strictness 跟标准的严格性

-aic 使能高级帧内编码 h263+

-umv 使能无限运动矢量 h263+

-deinterlace 不采用交织方法

-interlace 强迫交织法编码仅对mpeg2和mpeg4有效。当你的输入是交织的并且你想要保持交织以最小图像损失的时候采用该选项。可选的方法是不交织，但是损失更大

-psnr 计算压缩帧的psnr

-vstats 输出视频编码统计到vstats\_hhmmss.log

-vhook module 插入视频处理模块 module 包括了模块名和参数，用空格分开

D)音频选项

-ab bitrate 设置音频码率

-ar freq 设置音频采样率

-ac channels 设置通道 缺省为1

-an 不使能音频纪录

-acodec codec 使用codec编解码

E)音频/视频捕获选项

-vd device 设置视频捕获设备。比如/dev/video0

-vc channel 设置视频捕获通道 DV1394专用

-tvstd standard 设置电视标准 NTSC PAL(SECAM)

-dv1394 设置DV1394捕获

-av device 设置音频设备 比如/dev/dsp

F)高级选项

-map file:stream 设置输入流映射

-debug 打印特定调试信息

-benchmark 为基准测试加入时间

-hex 倾倒每一个输入包

-bitexact 仅使用位精确算法 用于编解码测试

-ps size 设置包大小，以bits为单位

-re 以本地帧频读数据，主要用于模拟捕获设备

-loop 循环输入流。只工作于图像流，用于ffserver测试

**二、Ffmpeg使用语法**

**Ffmpeg使用语法**

ffmpeg [[options][`-i' input\_file]]... {[options] output\_file}...

如果没有输入文件，那么视音频捕捉（只在Linux下有效，因为Linux下把音视频设备当作文件句柄来处理）就会起作用。作为通用的规则，选项一 般用于下一个特定的文件。如果你给 –b 64选项，改选会设置下一个视频速率。对于原始输入文件，格式选项可能是需要的。缺省情况下，ffmpeg试图尽可能的无损转换，采用与输入同样的音频视 频参数来输出。

ffmpeg转换所涉及到的选项较多，可参考[ffmpeg选项详解](http://www.ffmpeg.com.cn/index.php/Ffmpeg%E9%80%89%E9%A1%B9%E8%AF%A6%E8%A7%A3)。

ffmpeg支持多种文件格式和多种音频、视频编码器，可参考[ffmepg格式详解](http://www.ffmpeg.com.cn/index.php/Ffmepg%E6%A0%BC%E5%BC%8F%E8%AF%A6%E8%A7%A3)，（附：[常见视频文件格式详解](http://www.ffmpeg.com.cn/index.php/%E5%B8%B8%E8%A7%81%E8%A7%86%E9%A2%91%E6%96%87%E4%BB%B6%E6%A0%BC%E5%BC%8F%E8%AF%A6%E8%A7%A3)）

**视频文件截图**

* 截取一张352x240尺寸大小的，格式为jpg的图片

ffmpeg -i test.asf -y -f image2 -t 0.001 -s 352x240 a.jpg

* 把视频的前３０帧转换成一个Animated Gif

ffmpeg -i test.asf -vframes 30 -y -f gif a.gif

* 截取指定时间的缩微图

ffmpeg -i test.avi -y -f image2 -ss 8 -t 0.001 -s 350x240 test.jpg

-ss后跟的时间单位为秒

* 转换文件为3GP格式

ffmpeg -y -i test.mpeg -bitexact -vcodec h263 -b 128 -r 15 -s 176x144   
 -acodec aac -ac 2 -ar 22500 -ab 24 -f 3gp test.3gp

或

ffmpeg -y -i test.wmv -ac 1 -acodec libamr\_nb -ar 8000 -ab 12200 -s 176x144 -b 128 -r 15 test.3gp

**视频格式转换**

* 如何使用 ffmpeg 编码得到高质量的视频

**ffmpeg.exe** -i "D:\Video\Fearless\Fearless.avi" -target film-dvd -s 720x352  
 -padtop 64 -padbottom 64 -maxrate 7350000 -b 3700000 -sc\_threshold 1000000000   
 -trellis -cgop -g 12 -bf 2 -qblur 0.3 -qcomp 0.7 -me full -dc 10 -mbd 2  
 -aspect 16:9 -pass 2 -passlogfile "D:\Video\ffmpegencode" -an -f mpeg2video "D:\Fearless.m2v"

* 转换指定格式文件到**FLV**格式

**ffmpeg.exe** -i test.mp3 -ab 56 -ar 22050 -b 500 -r 15 -s 320x240 f:\test.**flv**   
**ffmpeg.exe** -i test.wmv -ab 56 -ar 22050 -b 500 -r 15 -s 320x240 f:\test.**flv**

* 转码解密的VOB

ffmpeg -i snatch\_1.vob -f avi -vcodec mpeg4 -b 800 -g 300 -bf 2 -acodec mp3 -ab 128 snatch.avi

上面的命令行将vob的文件**转化**成avi文件，mpeg4的视频和mp3的音频。注意命令中使用了B帧，所以mpeg4流是divx5兼容的。GOP大小是300意味着29.97帧频下每10秒就有INTRA帧。该映射在音频语言的DVD转码时候尤其有用。

* 同时编码到几种格式并且在输入流和输出流之间建立映射

ffmpeg -i /tmp/a.wav -ab 64 /tmp/a.mp2 -ab 128 /tmp/b.mp2 -map 0:0 -map 0:0

上面的命令行转换一个64Kbits 的a.wav到128kbits的a.mp2 ‘-map file:index’在输出流的顺序上定义了哪一路输入流是用于每一个输出流的。

* 转换文件为3GP格式

ffmpeg -i test.avi -y -b 20 -s sqcif -r 10 -acodec amr\_wb -ab 23.85 -ac 1 -ar 16000 test.3gp

注：如果要转换为3GP格式，则ffmpeg在编译时必须加上–enable-amr\_nb –enable-amr\_wb，详细内容可参考：[转换视频为3GPP格式](http://www.ffmpeg.com.cn/index.php/%E8%BD%AC%E6%8D%A2%E8%A7%86%E9%A2%91%E4%B8%BA3GPP%E6%A0%BC%E5%BC%8F)

**屏幕录制**

* 使用ffmpeg录像屏幕

ffmpeg -vcodec mpeg4 -b 1000 -r 10 -g 300 -vd x11:0,0 -s 1024x768 ~/test.avi

：其中，-vd x11:0,0 指录制所使用的偏移为 x=0 和 y=0，-s 1024×768 指录制视频的大小为 1024×768。录制的视频文件为 test.avi，将保存到用户主目录中

如果你只想录制一个应用程序窗口或者桌面上的一个固定区域，那么可以指定偏移位置和区域大小。使用xwininfo -frame命令可以完成查找上述参数。

* 重新调整视频尺寸大小

ffmpeg -vcodec mpeg4 -b 1000 -r 10 -g 300 -i ~/test.avi -s 800×600 ~/test-800-600.avi

注：ffmpeg的屏幕录制功能只能在Linux环境下有效。

**视频采集**

* 把摄像头的实时视频录制下来，存储为文件

ffmpeg -f video4linux -s 320\*240 -r 10 -i /dev/video0 test.asf

三、ffmepg使用

各种转换例子见：  
[FFmpeg Howto](http://www.itbroadcastanddigitalcinema.com/ffmpeg_howto.html)  
程序下载：  
[Unofficial FFmpeg Win32 Builds](http://arrozcru.no-ip.org/ffmpeg_builds/)(需要使用代理访问)  
或：  
[FFMPEG FULL SDK V3.0](http://www.bairuitech.com/html/ruanjianxiazai/20070812/51.html)

The generic syntax is:

ffmpeg [[infile options][`-i' infile]]… {[outfile options] outfile}…

As a general rule, options are applied to the next specified file. Therefore, order is important, and you can have the same option on the command line multiple times. Each occurrence is then applied to the next input or output file.

\* To set the video bitrate of the output file to 64kbit/s:  
  
ffmpeg -i input.avi -b 64k output.avi

\* To force the frame rate of the input and output file to 24 fps:

ffmpeg -r 24 -i input.avi output.avi

\* To force the frame rate of the output file to 24 fps:

ffmpeg -i input.avi -r 24 output.avi

\* To force the frame rate of input file to 1 fps and the output file to 24 fps:

ffmpeg -r 1 -i input.avi -r 24 output.avi

The format option may be needed for raw input files.

By default, FFmpeg tries to convert as losslessly as possible: It uses the same audio and video parameters for the outputs as the one specified for the inputs.

转换文件为3GP格式  
ffmpeg -y -i test.mpeg -bitexact -vcodec h263 -b 128 -r 15 -s 176x144 -acodec aac -ac 2 -ar 22500 -ab 24 -f 3gp test.3gp  
或  
ffmpeg -y -i test.mpeg -ac 1 -acodec amr\_nb -ar 8000 -s 176x144 -b 128 -r 15 test.3gp

转换指定格式文件到**FLV**格式

**ffmpeg.exe** -i test.mp3 -ab 56 -ar 22050 -b 500 -r 15 -s 320x240 f:\test.**flv**   
**ffmpeg.exe** -i test.wmv -ab 56 -ar 22050 -b 500 -r 15 -s 320x240 f:\test.**flv**  
\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  
ffmpeg -i F:\tools\ffmpeg.rev10464\test.avi -ab 56 -ar 22050 -b 500 -r 15 -s 320x240 F:\tools\ffmpeg.rev10464\test.**flv**  
ffmpeg -i "F:\tools\ffmpeg.rev10464\test.**flv**" -y -f image2 -ss 8 -t 0.001 -s 350x240 "test.jpg" 

**ffmpeg.exe** -i F:\闪客之家\闪客之歌.mp3 -ab 56 -ar 22050 -b 500 -r 15 -s 320x240 f:\11.**flv**   
ffmpeg -i F:\01.wmv -ab 56 -ar 22050 -b 500 -r 15 -s 320x240 f:\test.**flv**   
使用-ss参数 作用（time\_off set the start time offset），可以从指定时间点开始转换任务。如:   
转换文件格式的同时抓缩微图：   
ffmpeg -i "test.avi" -y -f image2 -ss 8 -t 0.001 -s 350x240 'test.jpg'   
对已有**flv**抓图：   
ffmpeg -i "test.**flv**" -y -f image2 -ss 8 -t 0.001 -s 350x240 'test.jpg'   
-ss后跟的时间单位为秒   
Ffmpeg转换命令   
ffmpeg -y -i test.mpeg -bitexact -vcodec h263 -b 128 -r 15 -s 176x144 -acodec aac -ac 2 -ar 22500   
-ab 24 -f 3gp test.3gp   
或者   
ffmpeg -y -i test.mpeg -ac 1 -acodec amr\_nb -ar 8000 -s 176x144 -b 128 -r 15 test.3gp   
  
  
ffmpeg参数设定解说   
-bitexact 使用标准比特率   
-vcodec xvid 使用xvid压缩   
-s 320x240 指定分辨率   
-r 29.97 桢速率（可以改，确认非标准桢率会导致音画不同步，所以只能设定为15或者29.97）   
画面部分，选其一   
-b <比特率> 指定压缩比特率，似乎ffmpeg是自动VBR的，指定了就大概是平均比特率，比如768，1500这样的   
就是原来默认项目中有的   
-qscale <数值> 以<数值>质量为基础的VBR，取值0.01-255，约小质量越好   
-qmin <数值> 设定最小质量，与-qmax（设定最大质量）共用，比如-qmin 10 -qmax 31   
-sameq 使用和源同样的质量   
声音部分   
-acodec aac 设定声音编码   
-ac <数值> 设定声道数，1就是单声道，2就是立体声，转换单声道的TVrip可以用1（节省一半容量），高品质   
的DVDrip就可以用2   
-ar <采样率> 设定声音采样率，PSP只认24000   
-ab <比特率> 设定声音比特率，前面-ac设为立体声时要以一半比特率来设置，比如192kbps的就设成96，转换   
君默认比特率都较小，要听到较高品质声音的话建议设到160kbps（80）以上   
-vol <百分比> 设定音量，某些DVDrip的AC3轨音量极小，转换时可以用这个提高音量，比如200就是原来的2倍   
这样，要得到一个高画质音质低容量的MP4的话，首先画面最好不要用固定比特率，而用VBR参数让程序自己去   
判断，而音质参数可以在原来的基础上提升一点，听起来要舒服很多，也不会太大（看情况调整   
  
  
例子：ffmpeg -y -i "1.avi" -title "Test" -vcodec xvid -s 368x208 -r 29.97 -b 1500 -acodec aac -ac 2 -ar 24000 -ab 128 -vol 200 -f psp -muxvb 768 "1.\*\*\*"   
  
解释：以上命令可以在Dos命令行中输入，也可以创建到批处理文件中运行。不过，前提是：要在ffmpeg所在的目录中执行（转换君所在目录下面的cores子目录）。   
参数：   
-y（覆盖输出文件，即如果1.\*\*\*文件已经存在的话，不经提示就覆盖掉了）   
-i "1.avi"（输入文件是和ffmpeg在同一目录下的1.avi文件，可以自己加路径，改名字）   
-title "Test"（在PSP中显示的影片的标题）   
-vcodec xvid（使用XVID编码压缩视频，**不能**改的）   
-s 368x208（输出的分辨率为368x208，注意片源一定要是16:9的不然会变形）   
-r 29.97（帧数，一般就用这个吧）   
-b 1500（视频数据流量，用-b xxxx的指令则使用固定码率，数字随便改，1500以上没效果；还可以用动态码率如：-qscale 4和-qscale 6，4的质量比6高）   
-acodec aac（音频编码用AAC）   
-ac 2（声道数1或2）   
-ar 24000（声音的采样频率，好像PSP只能支持24000Hz）   
-ab 128（音频数据流量，一般选择32、64、96、128）   
-vol 200（200%的音量，自己改）   
-f psp（输出psp专用格式）   
-muxvb 768（好像是给PSP机器识别的码率，一般选择384、512和768，我改成1500，PSP就说文件损坏了）   
"1.\*\*\*"（输出文件名，也可以加路径改文件名）   
  
机器强劲的话，可以多开几个批处理文件，让它们并行处理。   
E:\**ffmpeg.exe** -i I:\1.wmv -b 360 -r 25 -s 320x240 -hq -deinterlace -ab 56 -ar 22050 -ac 1 D:\2.**flv**   
===========================================  
**ffmpeg.exe** -i F:\闪客之家\闪客之歌.mp3 -ab 56 -ar 22050 -b 500 -r 15 -s 320x240 f:\11.**flv** ffmpeg -i F:\01.wmv -ab 56 -ar 22050 -b 500 -r 15 -s 320x240 f:\test.**flv** 使用-ss参数 作用（time\_off set the start time offset），可以从指定时间点开始转换任务。如:  
转换文件格式的同时抓缩微图：  
ffmpeg -i "test.avi" -y -f image2 -ss 8 -t 0.001 -s 350x240 'test.jpg'  
对已有**flv**抓图：  
ffmpeg -i "test.**flv**" -y -f image2 -ss 8 -t 0.001 -s 350x240 'test.jpg'  
-ss后跟的时间单位为秒 Ffmpeg转换命令  
ffmpeg -y -i test.mpeg -bitexact -vcodec h263 -b 128 -r 15 -s 176x144 -acodec aac -ac 2 -ar 22500 -ab 24 -f 3gp test.3gp  
或者  
ffmpeg -y -i test.mpeg -ac 1 -acodec amr\_nb -ar 8000 -s 176x144 -b 128 -r 15 test.3gp ffmpeg参数设定解说   
-bitexact 使用标准比特率   
-vcodec xvid 使用xvid压缩   
-s 320x240 指定分辨率   
-r 29.97 桢速率（可以改，确认非标准桢率会导致音画不同步，所以只能设定为15或者29.97）   
  
  
画面部分，选其一   
-b <比特率> 指定压缩比特率，似乎ffmpeg是自动VBR的，指定了就大概是平均比特率，比如768，1500这样的就是原来默认项目中有的   
-qscale <数值> 以<数值>质量为基础的VBR，取值0.01-255，约小质量越好   
-qmin <数值> 设定最小质量，与-qmax（设定最大质量）共用，比如-qmin 10 -qmax 31   
-sameq 使用和源同样的质量 声音部分   
-acodec aac 设定声音编码   
-ac <数值> 设定声道数，1就是单声道，2就是立体声，转换单声道的TVrip可以用1（节省一半容量），高品质的DVDrip就可以用2   
-ar <采样率> 设定声音采样率，PSP只认24000   
-ab <比特率> 设定声音比特率，前面-ac设为立体声时要以一半比特率来设置，比如192kbps的就设成96，转换君默认比特率都较小，要听到较高品质声音的话建议设到160kbps（80）以上   
-vol <百分比> 设定音量，某些DVDrip的AC3轨音量极小，转换时可以用这个提高音量，比如200就是原来的2倍 这样，要得到一个高画质音质低容量的MP4的话，首先画面最好不要用固定比特率，而用VBR参数让程序自己去判断，而音质参数可以在原来的基础上提升一点，听起来要舒服很多，也不会太大（看情况调整 例子：ffmpeg -y -i "1.avi" -title "Test" -vcodec xvid -s 368x208 -r 29.97 -b 1500 -acodec aac -ac 2 -ar 24000 -ab 128 -vol 200 -f psp -muxvb 768 "1.\*\*\*"  
  
解释：以上命令可以在Dos命令行中输入，也可以创建到批处理文件中运行。不过，前提是：要在ffmpeg所在的目录中执行（转换君所在目录下面的cores子目录）。  
参数：  
-y（覆盖输出文件，即如果1.\*\*\*文件已经存在的话，不经提示就覆盖掉了）  
-i "1.avi"（输入文件是和ffmpeg在同一目录下的1.avi文件，可以自己加路径，改名字）  
-title "Test"（在PSP中显示的影片的标题）  
-vcodec xvid（使用XVID编码压缩视频，**不能**改的）  
-s 368x208（输出的分辨率为368x208，注意片源一定要是16:9的不然会变形）  
-r 29.97（帧数，一般就用这个吧）  
-b 1500（视频数据流量，用-b xxxx的指令则使用固定码率，数字随便改，1500以上没效果；还可以用动态码率如：-qscale 4和-qscale 6，4的质量比6高）  
-acodec aac（音频编码用AAC）  
-ac 2（声道数1或2）  
-ar 24000（声音的采样频率，好像PSP只能支持24000Hz）  
-ab 128（音频数据流量，一般选择32、64、96、128）  
-vol 200（200%的音量，自己改）  
-f psp（输出psp专用格式）  
-muxvb 768（好像是给PSP机器识别的码率，一般选择384、512和768，我改成1500，PSP就说文件损坏了）  
"1.\*\*\*"（输出文件名，也可以加路径改文件名）  
  
P.S. 版主机器强劲的话，可以多开几个批处理文件，让它们并行处理。 E:\**ffmpeg.exe** -i I:\1.wmv -b 360 -r 25 -s 320x240 -hq -deinterlace -ab 56 -ar 22050 -ac

附 关于FFmpeg 其他文章

http://www.ssyleo.com/ffmpeg.html